AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application.

Listing of claims:

Please cancel claim 3 without prejudice.

1. (currently amended) A method for managing the processing of commands issued on a bus via a plurality of agents, comprising:

allocating ownership of the bus to the plurality of agents based on a predetermined bus arbitration order;

queuing agent identifiers of the agents that issued commands which were retried due to unavailable processing resources; and

granting command processing priority of the processing resources to agents corresponding to the queued agent identifiers relative to other agents having ownership of the bus in an order in which the queued agent identifiers were queued relative to each other~~accordance with the predetermined bus arbitration order~~.

2. (original) The method of Claim 1, further comprising issuing retry responses for commands issued by the other agents to avail the processing resources to the agents corresponding to the queued agent identifiers.

3. (canceled)

4. (original) The method of Claim 1, wherein queuing agent identifiers comprises queuing the agent identifiers in a first-in, first-out fashion.

5. (original) The method of Claim 4, wherein granting command processing priority of the processing resources comprises granting command processing priority to the agents corresponding to the queued agent identifiers in the first-in, first-out fashion.

6. (original) The method of Claim 1, wherein granting command processing priority of the processing resources comprises always granting command processing priority to the agents corresponding to the queued agent identifiers relative to the other agents having ownership of the bus.

7. (original) The method of Claim 1, wherein granting command processing priority of the processing resources comprises granting command processing priority to the agents corresponding to the queued agent identifiers relative to the other agents having ownership of the bus in accordance with a predefined fairness algorithm.

8. (original) The method of Claim 1, further comprising regulating the command processing priority between the agents corresponding to the queued agent identifiers and the other agents having ownership of the bus.

9. (original) The method of Claim 8, wherein regulating the command processing priority comprises selectively granting the command processing priority of the processing resources to agents corresponding to the queued agent identifiers relative to the other agents having ownership of the bus.

10. (original) The method of Claim 9, wherein selectively granting the command processing priority comprises disregarding the granting of the command processing priority to the agents corresponding to the queued agent identifiers relative to the other agents having ownership of the bus in accordance with a configurable pattern.

11. (original) The method of Claim 1, further comprising imposing queue entry rules on the agent identifiers of the agents that issued commands which were retried, and wherein queuing the agent identifiers comprises queuing only those agent identifiers complying with the queue entry rules.

12. (original) The method of Claim 11, wherein imposing queue entry rules comprises prohibiting queuing the agent identifiers associated with commands that were retried if the agent identifier has already been queued.

13. (original) The method of Claim 11, wherein imposing queue entry rules comprises prohibiting queuing the agent identifiers associated with commands that were retried if the processing resources are currently processing a command for the corresponding agent.

14. (previously presented) A method for managing the processing of commands issued on a bus via a plurality of agents, comprising:

allocating bus ownership to the plurality of agents based on a predetermined bus arbitration cycle;

determining whether processing resources are available to process the commands issued by the agents that have been granted bus ownership;

initiating a retry for the commands that were not processed due to unavailable processing resources;

queuing agent identifiers corresponding to each of the agents in which a retry was initiated;

receiving one or more controllable configuration parameters indicative of a desired processing ratio in which the commands issued according to the predetermined bus arbitration cycle are to be processed relative to the commands associated with the queued agent identifiers; and

overriding the predetermined bus arbitration cycle and processing the commands associated with the agent identifiers that have been queued according to the desired processing ratio indicated by the controllable configuration parameters.

15. (canceled)

16. (canceled)

17. (previously presented) The method of Claim 14, wherein the one or more controllable configuration parameters comprise a count representing a number of times in which the processing of the queued agent identifiers will be not be allowed to override the predetermined bus arbitration cycle.

18. (original) The method of Claim 14, wherein overriding the predetermined bus arbitration cycle comprises allowing processing of the commands associated with the queued agent identifiers vis-à-vis the commands issued on the bus in accordance with the predetermined bus arbitration cycle.

19. (original) The method of Claim 14, wherein initiating a retry for the commands that were not processed comprises issuing a retry response in response to determining that the processing resources are unavailable.

20. (original) The method of Claim 14, further comprising imposing queue entry rules on the agent identifiers corresponding to the agents in which a retry was initiated, and wherein queuing the agent identifiers comprises queuing only those agent identifiers complying with the queue entry rules.

21. (original) The method of Claim 14, wherein the processing resources comprise at least one read thread.

22. (original) The method of Claim 14, wherein the processing resources comprise at least one write thread.

23. (previously presented) A method for managing the processing of commands issued on a bus via a plurality of agents, comprising:

allocating ownership of the bus to the plurality of agents based on a predetermined bus arbitration order;

queuing agent identifiers identifying the agents that issued commands which were retried due to unavailable processing resources;

granting command processing priority of the processing resources to agents corresponding to the queued agent identifiers relative to other agents having ownership of the bus in accordance with the predetermined bus arbitration order; and

dynamically controlling a frequency in which the command processing is granted priority to the agents corresponding to the queued agent identifiers relative to the other agents having ownership of the bus.

24. (previously presented) The method of Claim 23, wherein dynamically controlling a frequency comprises receiving one or more controllable configuration parameters, and regulating the frequency at which the queued agent identifiers are granted command processing priority in response to the controllable configuration parameters.

25. (original) The method of Claim 24, wherein regulating the frequency at which the queued agent identifiers are granted command processing priority comprises enabling the queued agent identifiers to be output from a queue for processing after a predetermined number of valid queued agent identifiers have been passed over for processing.

26. (original) The method of Claim 25, wherein enabling the queued agent identifiers to be output from a queue for processing comprises:

comparing a predetermined pass over count to a current pass over count;

incrementing the current pass over count each time a valid queued agent identifier has been passed over for processing; and

enabling the queued agent identifiers to be output from the queue when the current pass over count reaches the predetermined pass over count.

27. (previously presented) A cooperative arbitration and processing resource allocation system, comprising:

(a) an established-order arbiter for allocating bus ownership among a plurality of bus agents;

(b) a processing fairness module, comprising:

(i) a queue to store bus agent identifiers (IDs) corresponding to bus agents that issued commands which were subjected to a retry due to unavailable processing resources;

(ii) a dynamically configurable queue output throttle to adjust an availability of a valid bus agent ID at an output of the queue;

(c) an override module coupled to the established-order arbiter and the processing fairness module to grant command processing priority of the processing resources to either the bus agents corresponding to valid bus agent IDs or to the bus agents that have been granted bus ownership, depending on whether the valid bus agent ID at the output of the queue is available as determined by the queue output throttle.

28. (previously presented) The cooperative arbitration and processing resource allocation system as in Claim 27, wherein the dynamically configurable_queue output throttle comprises:

a maximum starvation register to store a predetermined number of valid bus agent IDs that will be passed over for processing;

a current starvation register to store a current number of valid bus agent IDs that have been passed over for processing; and

a compare module coupled to the maximum starvation register and the current starvation register to compare the current and predetermined numbers of valid bus agent IDs that have been passed over for processing, and to output an availability indicator to indicate the availability of the valid bus agent ID.

29. (original) The cooperative arbitration and processing resource allocation system as in Claim 28, further comprising an incrementing module coupled to the current starvation register to increment the current starvation register each time that the valid bus agent ID has been passed over for processing.

30. (original) The cooperative arbitration and processing resource allocation system as in Claim 29, further comprising a reset module coupled to the current starvation register to reset the current starvation register upon the output of the availability indicator.

31. (original) The cooperative arbitration and processing resource allocation system as in Claim 28, wherein the maximum starvation register comprises an input to receive the predetermined number of valid bus agent IDs that will be passed over for processing.

32. (original) The cooperative arbitration and processing resource allocation system as in Claim 27, further comprising a fairness entry validation module to prohibit entry of the bus agent IDs onto the queue that do not comply with one or more predetermined queue entry rules.

33. (previously presented) A system for allocating command processing resources, comprising:

(a) a plurality of bus agents each capable of issuing commands;

(b) an input/output (I/O) bus coupled to each of the bus agents to facilitate transfer of the commands;

(c) a memory bus to facilitate transfer of the commands to and from a memory; and

(d) an I/O bridge module to interface the I/O bus and the memory bus, wherein the I/O bridge module comprises:

(1) an established-order arbiter for allocating bus ownership among a plurality of bus agents;

(2) a processing fairness module, comprising:

(i) a queue to store bus agent identifiers (IDs) corresponding to bus agents that issued commands which were subjected to a retry due to unavailable processing resources;

(ii) a dynamically configurable_queue output throttle to adjust an availability of a valid bus agent ID at an output of the queue; and

(3) an override module coupled to the established-order arbiter and the processing fairness module to grant command processing priority of the processing resources to either the bus agents corresponding to valid bus agent IDs or to the bus agents that have been granted bus ownership, depending on whether the valid bus agent ID at the output of the queue is available as determined by the queue output throttle.

34. (original) The system as in Claim 33, further comprising a plurality of the I/O buses, each coupled to a different plurality of the bus agents, and wherein the I/O bridge module comprises an established-order arbiter, a processing fairness module, and an override module for each of the plurality of the I/O buses.

35. (canceled)

36. (previously presented) A cooperative arbitration and processing resource allocation system for managing the processing of commands issued on a bus via a plurality of agents, comprising:

means for allocating ownership of the bus to the plurality of agents based on a predetermined bus arbitration order;

means for queuing agent identifiers of the agents that issued commands which were retried due to unavailable processing resources;

means for granting command processing priority of the processing resources to agents corresponding to the queued agent identifiers relative to other agents having ownership of the bus in accordance with the predetermined bus arbitration order; and

means for controlling a frequency in which the command processing is granted priority to the agents corresponding to the queued agent identifiers relative to the other agents having ownership of the bus.